PROBABILISTIC SEARCH: A BAYESIAN APPROACH IN A CONTINUOUS WORKSPACE

BY

DEVIN ASHISH BONNIE

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Seth Andrew Hutchinson

ABSTRACT

This thesis considers the problem of modeling search for a single, non-moving target in a continuous environment, where the search agent's only observations are obtained from a binary sensor. To model this problem, the widely used Bayesian filtering approach is employed to obtain the general filtering equations for the posterior distribution representing the object's location over the workspace. Given a likelihood and prior belief belonging to the exponential family class, while using this class's self-conjugacy property, an exact, finite representation of the object posterior is explicitly derived. Though complexity issues may render this exact representation infeasible for computation, regularized particle filtering is utilized to yield a continuous approximation of the object belief. To demonstrate the validity of the search model, a gradient-ascent search strategy is applied with care taken to avoid local maxima. This is done with multiple simulations for various prior distributions. Finally, future work is described for search applications and approximation schemas relevant to the structure of the search model presented.

To my friends and family for their unconditional support

ACKNOWLEDGMENTS

I would like to thank my adviser Professor Seth Hutchinson for his insight and guidance throughout the course of my graduate study. I would also like to thank my fellow research group member Salvatore Candido not only for his collaboration, but for his academic camaraderie as well.

TABLE OF CONTENTS

1 INTRODUCTION

The problem of search has been well known for quite some time. For a lost object, humans intuitively consider hypotheses to engage some search routine to locate lost items, e.g. a wallet or car keys. Here, the importance of such a subject can be easily inferred. Rescue missions for lost persons [1], ships at sea [2,3], and even the location of misplaced or discarded weapons [4,5] have been subjects of interest in the formulation of the search problem. Rather than modeling hypotheses concerning an object in a deterministic fashion, the accepted, standard approach uses a probabilistic representation. The key advantage of representing the search process probabilistically is its appropriate modeling of the possible multiple hypotheses for an object's location. These hypotheses are modeled as probability distributions over the area for which the search is conducted. Such a form encapsulates the uncertainty associated with search and the stochastic nature of our world. Thus, it is important to model the search problem efficiently in order to proceed with a solution.

The origin of search theory arose out of the US Naval Operations Research community [6] with the purpose of efficiently detecting submarines during the second World War. This initial mathematical formalization characterized the problem with a prior probability distribution, detection probability, and the density of search effort. The prior represents all initial information known at the time before any search effort has begun, where the prior is strictly in the form of a probability distribution. Figure 1.1 illustrates an example prior for a discrete environment. Information gathered during the process, e.g. radio transmissions, radar, and other various sensing instruments, forms the basis of the detection probability.

To characterize the probability of detection, conditional probabilities are formulated in conjunction with the gathered information, e.g. conditioning the sensed information upon the position of (or other properties describing) the searcher. Once the prior and detection probabilities are obtained, Bayes'

theorem is used to obtain a new, updated probability distribution known as the posterior. The posterior represents the new probability distribution incorporating the sensed information. From this representation it is desired to maximize the likelihood of target detection by placing constraints on the search (or allocation of) effort, e.g. the time spent in a given area or energy depleted. By constraining this problem, optimality can now be defined.

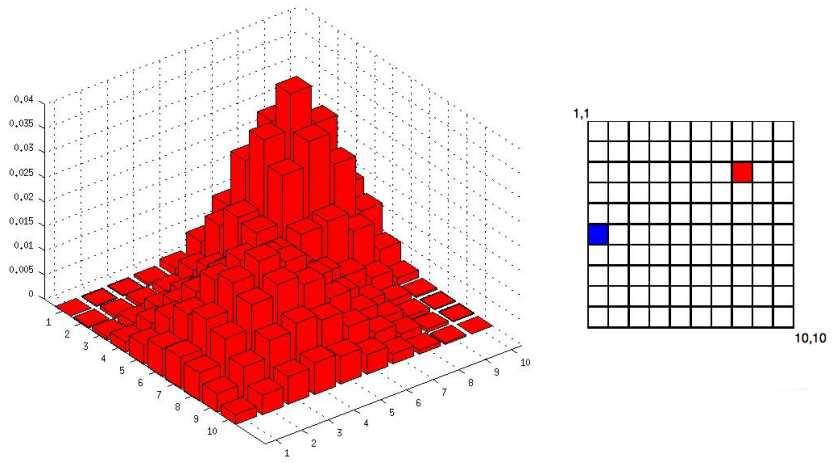


Figure 1.1: Example of a prior distribution for a discrete search model. The left plot denotes the probability mass function, where each cell is part of its support. The right plot represents the position of the search agent (blue cell) and the true position of the object (red cell).

Optimal search [4] maximizes the probability of target detection within some limit given by a cost or imposed constraint. Constraining the problem allows the characterization of limited resources available to the searcher. These constraints intuitively led to Lagrange multiplier techniques, yielding sufficient and necessary conditions for the constrained maximization of detection. Likewise, the formulation of two different types of optimality, namely *uniformly optimal* and *locally optimal*, arose from said constraints. Uniformly optimal search constrains the problem for a given search effort and maximizes the probability of detection. To obtain such a search plan, a regular detection function, i.e. a likelihood being positive, continuous, and strictly decreasing derivative, is necessary. Local optimality minimizes the expected cost of finding the target, given some cost function. Though posed separately, [4] shows that the two optimal search formulations are equivalent.

A vast amount of literature has been dedicated to the one-sided search problem; that is, the target does not respond to the searcher and is stationary

or possibly moving [7, 8]. Much of the work pertained to the approximation of optimal plans. Again, the approaches taken utilized the previous Lagrange multiplier method discussed. However, most of the search areas considered were discrete, thus limiting discussion of distributions and their evolutions to probability mass functions. Other work represented the search area as a discrete space with the intent to again maximize probability of detection or minimize the expected cost of search, with respect to measures as search effort or time.

Defining particular models of search led to the usage of Markov chains to describe target trajectories yielding an efficient method for Bayesian updating. From this set of work, variations of the one-sided problem were developed for moving targets, evading targets, and miscellaneous "games." Earlier work in [9] analyzes coordinated and random strategies for multiple search agents with imperfect (probabilistic) sensor models. Tradeoffs between the two strategies provide insight as the sensor model accuracy is varied, i.e. it is better to randomly search the target area given a very noisy sensor.

This concludes the discussion of the origins and early methods of the search problem. The thesis continues by describing problems similar to probabilistic search and providing a brief overview. Specifically, these formulations have been chosen as their work maintains the Bayesian formulation illustrated above.

1.1 Related Areas

Various fields of robotics research are related to probabilistic search. This diverse list includes exploration and mapping, navigation, localization, target tracking, and area coverage. Though spanning such a wide area, these topics are unified by their Bayesian approach by employing observations to improve planning trajectories in an online fashion while integrating sensor data with the problem specific model.

Simultaneous localization and mapping (SLAM) [10] is a solution to provide autonomy to mobile platforms. A robot is placed in an unknown environment with an unknown location and given the goal to incrementally, accurately construct a map while simultaneously localizing itself. Much of the SLAM development was achieved through the use of the Kalman filter [11],

which utilizes the Bayesian filter to predict the current state estimate and to incorporate sensor measurements. However, using the Kalman filter imposes limitations, namely the inability to represent complex feature models or environments [12]. Thus, the SLAM problem is posed in a more general Bayesian estimation formulation. This is accomplished by initializing the prior as a joint probability distribution (specifically a mixture of Gaussians) for the robot's state and location of a landmark conditioned on all previous landmark observations, the previous control inputs to the robot, and the initial robot state. Observations of landmarks are modeled as the conditional probability of observing a landmark given the current state and landmark location, defining the likelihood. Finally the state transition of the robot is assumed to be Markovian, i.e. the next state only depends on the preceding state and its information (control input) rather than the entire history. With these components, the recursive Bayesian updates occur in a two-step process. The first updates the robot's position distribution, given the previous state and control history, by the Chapman-Komorgorov equation [13]. This first step is referred to as the prediction, or time-update. Then this distribution is fused with the observation model by applying Bayes' theorem given the observation likelihood, which completes the measurement update step. Furthermore, the Bayesian recursive process and Markovian motion assumptions are also utilized for localization [14] and navigation [15].

Similarly, target tracking techniques for autonomous agents utilize the recursive Bayesian update in order to sufficiently model the uncertainty associated with sensing models and target motion [16]. Likewise, active visual search employs target priors and the canonical Bayesian update, as discussed previously in the introduction, given an observation model. Again, it is desirable to yield an optimal solution by minimizing the expected cost of search. This is illustrated in the experimental setting of service robots searching for specific objects is various rooms [17]. Additionally, target classification is modeled in [18] with object likelihoods (Gaussian) and probabilistic camera models, i.e. probability distributions given the current camera parameters. After each observation (positive or negative), the target state is updated via Bayes' theorem. Another recent tracking approach [19] models the initial search process by using the recursive Bayesian update, which generates individual target distributions fusing multiple agent observations. Search is performed until a target has been detected and tracking occurs upon detec-

tion.

Area coverage [20] approaches the problem of a robot passing over all points in a given space. By partitioning this space, known as *cellular decomposition* [14], it is possible to assert the completeness of a coverage plan. Cellular decomposition divides the desired space to be covered such that the cells provide a simple structure to plan a coverage path. Once a cell is visited, the robot can then cover the interior of said cell with basic motions, thus exhaustively covering a subset of the desired points. Furthermore, such planning can be accomplished by utilizing a probabilistic map to generate the optimal path for maximizing object detection [5] and acts as an extension to classical motion planning methods. By generating a posterior distribution of mine locations, with a Bayesian approach capturing the uncertainty of location, efficient de-mining is possible.

This concludes the overview of related work. The thesis continues by presenting current work in probabilistic search. Specifically, these search models are emphasized rather than the search technique chosen for object detection, as the majority of this thesis pertains to the problem of modeling the search problem.

1.2 Recent Work

From its origins in operations research, the model of search, in recent works, retains its Bayesian flavor. Both discrete and continuous search spaces are still studied in depth, while examining the tradeoffs of various search strategies. Though much of this work focuses on the actual optimization or search method used, this thesis concentrates on the search model employed.

Continuous models of search [21, 22] usually employ the standard recursive Bayesian procedure. Similarly to the SLAM problem, the Markov assumption is taken into account, encapsulating the possibility of a non-stationary target. Sensor observations are taken at unique time intervals, where the usage of the standard update step using Bayes' theorem occurs after each observation. Radar sensors are employed for both the single and multiple target problems where the likelihood of detection is a function of received power and the signal-to-noise ratio [21]. For multiple targets, the notion of independence is again utilized, which asserts that a target position does not

affect any other in the search space because they are unrelated. This yields separate distributions for each target, where the update step propagates new information individually. Such an assertion, for separate target distributions, is intuitive and allows any well-modeled search problem to accommodate the possibility of multiple targets. However, these continuous search models are left in a loose, generalized form. Usually these models are left in a general form citing the case of complex posterior evolution; but, for implementation, Gaussian prior densities are typically used to represent the target distributions.

A discrete search model dissects a continuous environment into disjoint cells, similar to grid-world environments. The entire search area, i.e. every cell, is the support of the probability mass function that describes the target prior (Figure 1.1). In this case it is assumed that the target is actually within the given search area. However, in the case of possible target absence, the discrete model utilizes a "virtual" cell [23] that considers all the other area outside the given search environment. Again, this cell is assigned a prior probability measure. An advantage of using a discrete environment is the ease of producing a closed-form, rather than a generalized Bayesian, expression for the target probability evolution that is readily extensible toward the use of multiple search agents [24, 25]. This can be done not only for a single-time interval because, given the set of progressive observations, each cell's evolution can be independently observed; but due to the nature of the model, only uniform or (obviously) discrete priors can be considered. Concurrently, this model represents a coarse view of the search area and a finer representation, though more expressive, yields higher model complexity, e.g. 100 cells vs $100,000$ cells. One such discrete representation, recently applied towards search, that allows greater expressiveness is a quadtree [26].

Observations, given simply as probabilities of detection or no detection, are used commonly throughout these works. Some of the more complex sensor likelihoods used are radar waves [22], cameras (as discussed in related work), or signal strength models. Inversely, simpler models are implemented such as ones that represent detection of the target in a discrete cell with a Bernoulli distribution [23]. An advantage of modeling a Bernoulli sensor likelihood is it yields the least amount of information provided by a sensor, e.g. binary. A simple example is a search game played by children. When searching for an object hidden by another person, the searcher asks for information. The

one with knowledge of the object responds with "hot" or "cold," depending on the proximity of the searcher to the target. From this, the person searching adjusts their plan (and belief of the object position) and continues the process. Thus, search is represented in its simplest form: observation with a binary sensor.

This concludes the introduction of the thesis. Chapter 2 presents the search problem in the Bayesian framework and describes its necessary preliminaries. Chapter 3 models and discusses the target hypothesis and gives an explicit representation of the target evolution. Then, an approximation method to reduce the complexity of the exact representation is presented. Chapter 4 presents the results for the search methods illustrated. Finally, Chapter 5 concludes the thesis and describes possible directions for future work.

2 THE SEARCH PROBLEM

This chapter presents the formalized problem for the Bayesian search approach. The definitions for the robot's motion model, its environment, and the problem state are given. These definitions are utilized within the Bayesian filtering equations, which are shown explicitly.

2.1 Formalization

The search agent (or robot) conducts a search for a stationary object with an unknown position in a continuous workspace. A search strategy is halted once the robot "finds" the object, i.e. the robot occupies the position where the object is located. The workspace, denoted as $\mathcal{W}$, considered is limited to $\mathbb{R}^1$ and $\mathbb{R}^2$, but the framework given may be generalized to $\mathbb{R}^n$. Here the robot state is represented as

$$x^r \in \mathcal{W}$$

and has the following discretized kinematic motion model

$$x^r_{t+1} = x^r_t + u_t, \qquad ||u_t|| \leq \bar{u} \tag{2.1}$$

where the subscript t denotes the time-step. The quantity u_t is the control input to the system that is bounded by some $\bar{u}$ representing the robot's motion constraints. This yields the control space, specifically because this space is simply the control sent to the robot, depending on the given $\mathcal{W}$.

A brief introduction to the sensor model is discussed here, whereas the model specifics are fully presented in Section 3.2. The robot is equipped with a probabilistic binary sensor [19,23,24], which produces the observation space of $\{1, 0\}$, where one and zero represent detection and no detection, respectively. That is, for some x^r_t an observation is taken where a "one" indicates that the object is close to the robot and "0" otherwise. Note that

this model does not account for the possibilities of false positives or negatives as given in recent work presented in Section 1.2. However, the usage of a binary sensor yields the least amount of information of the object's state in the search space. Thus, this thesis models the search process not only probabilistically but with minimal knowledge. The observation at time t is denoted as $\boldsymbol{y}_t$, where the bold symbol indicates a random variable. As the robot approaches the object, the observation of $\boldsymbol{y}_t = 1$ becomes more probable. Furthermore, at time-step t, the observation of $\boldsymbol{y}_t$ is conditionally independent of the robot's control, position, sensor history, and the position of the object when it is conditioned on the current robot position and representation of the object position. This conditional independence assumption is given with the specifics of the sensor model. Again, the location of the object is not known to the robot, but it is represented as a probability distribution.

As the state x_t^r parameterizes the configuration of the robot, the problem state is presented which contains the necessary parameters for the search model. The problem state is defined as the pair

$$x_t := (x_t^r, \boldsymbol{x}_t^o)$$

where $\boldsymbol{x^o}$ is a random variable that represents the position of the object in $\mathcal{W}$. This random variable is distributed according to a finitely parameterizable probability density function (PDF), designated as $\mathrm{f}_{\boldsymbol{x^o}}(x^o)$. The PDF represents the prior information of the object's location over $\mathcal{W}$ (see Figure 2.1), which can be termed as the *target distribution* [4]. Note that this representation must satisfy the usual properties of probability density functions: its cumulative density function must be continuous and the integral of said PDF, over its support, must be equal to one [27]. It is necessary to model the object's location as a probability density in order to sufficiently realize the search problem, since the object state is not directly observable as a direct consequence of the given sensor model. The robot's kinematic motion model considered in Equation 2.1 is deterministic, i.e. the state of the robot is known. Such a proposal is not unusual [4, 19, 21, 24] because this thesis pertains to modeling the propagation of the object's uncertainty. The uncertainty inherent in the problem is generated from the unknown initial condition, namely the location of the object.

The advantage of using a probability density function is intuitive as it

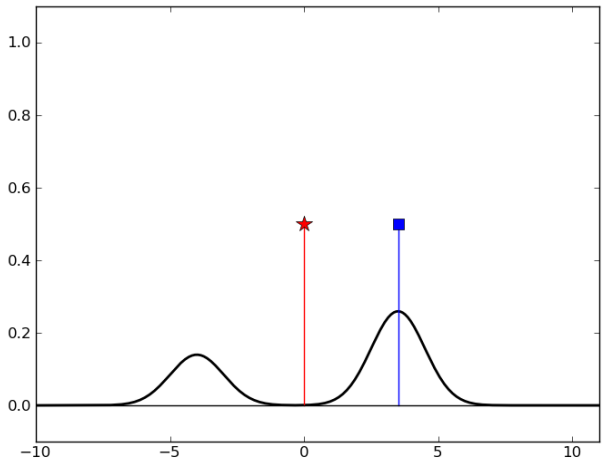
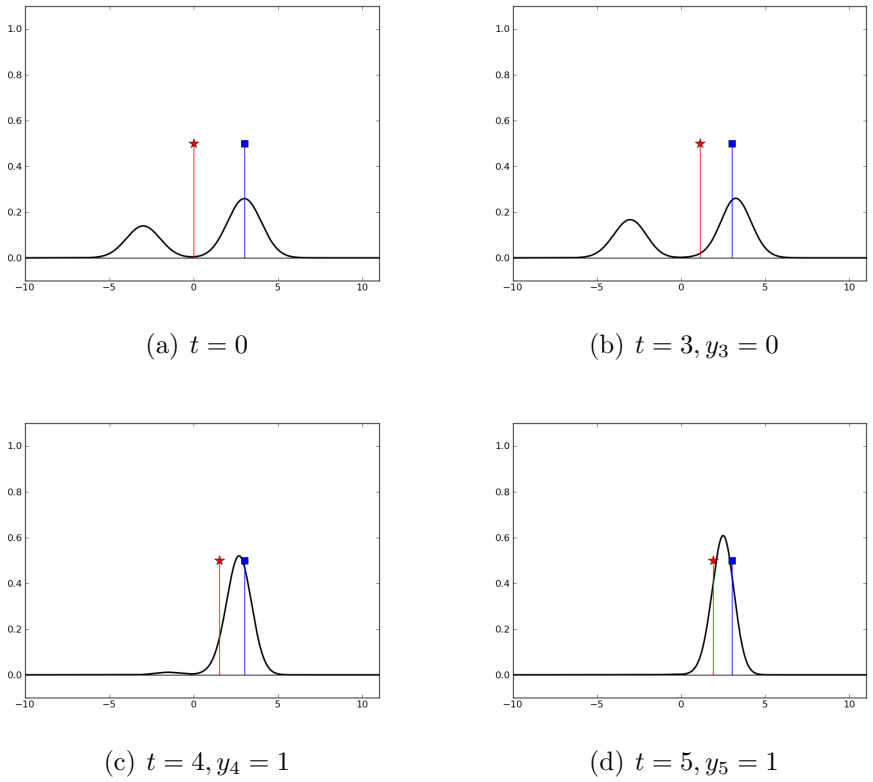


Figure 2.1: Example of a target distribution. The black line represents the target distribution for $\mathcal{W} = \mathbb{R}^1$. The red impluse with a star indicates the robot's position while the blue impulse with a square denotes the object's true location (unknown to the robot).

yields a continuous representation of the probability of target location. This in turn allows the usage of gradient information to define a control law for search. Such a control law allows the searcher to follow a gradient-ascent strategy, i.e. move in $\mathcal{W}$ towards a local maximum of the belief. As shown in [21, 24] this strategy is sub-optimal but desirable as the optimal solution is highly complex or intractable. As this strategy yields the control for the robot, sensor measurements are taken at each time-step, i.e. after each movement when the control is applied. If measurements were not taken along such a path given by the strategy [23], the evolution of the target belief is halted until more sensor information is accrued. This is far from desirable as less information is available towards eliminating "false" hypotheses, e.g. the robot moves towards a hypothesis away from the true position of the object and does not update the map until a peak of the belief is reached. Thus sensor observations will be taken at every time-step. Given a continuous representation of the object map, a greedy strategy follows the direction of the positive gradient, i.e. the desired robot state $\tilde{x}^r_{t+1}$ at time $t+1$ is

$$\tilde{x}^r_{t+1} = x^r_t + \gamma \, \nabla \mathrm{f}_{\boldsymbol{x}_o}\left(x^r_t\right) \tag{2.2}$$



(a) $t = 0$ (b) $t = 3, y_3 = 0$

(c) $t = 4, y_4 = 1$ (d) $t = 5, y_5 = 1$

Figure 2.2: Evolution of the target belief for $\mathcal{W} = \mathbb{R}^1$ using observations from a binary sensor (time-steps $t = 1$ and $t = 2$ omitted, both with observations of $y_t = 0$). Note how, over time, the maximum of the rightmost component of the belief does not correspond to the object's true position. As the robot moves towards the maximum and observes positive detections, the belief shifts towards the robot.

for a sufficiently small γ. The control input u_t is then given as

$$u_t = \tilde{x}^r_{t+1} - x^r_t \tag{2.3}$$

such that the control input satisfies the constraint given in Equation 2.1. However, when employing gradient methods, it is imperative that care is taken for the case that local maxima (or minima for gradient descent) exist. For this search model, as it is explained in Chapter 3, such a case is possible when the robot approaches the true object position or if the prior belief does not accurately reflect the object's true position. Figure 2.2 illustrates such a case where the peak of the belief shifts from the object's true position towards the robot's current position in the workspace, given positive sensor

observations. As the robot observes negative detections, i.e. $\boldsymbol{y}_t = 0$, the corresponding probability is shifted from the area closest to the robot to the more likely areas in the target belief (subfigures a and b). When approaching the true object position and observing positive detections, i.e. $\boldsymbol{y}_t = 1$, the area closest to the robot is scaled up, which asserts the hypothesis that the object is indeed nearby. The result is that the peak of the belief is shifted towards the robot's current position (subfigures c and d). Hence, by following the gradient ascent strategy, the robot will reach a steady-state at some x_t^r, given successive positive detections, but will not find the object.

To overcome this drawback, a method used in gradient following problems will be employed. Similar to escaping minima in potential fields [28, 29], a random walk effectively perturbs the robot from the steady-state. The method used is quite straightforward: for $\mathcal{W} = \mathbb{R}^2$ and given $\bar{u}$, an angle (ϕ) is drawn from a uniform distribution with support from 0 to 360°. Then the robot moves accordingly given the direction ϕ and magnitude $\bar{u}$ from the steady-state. This random movement is repeated for a number of time-steps in order to escape the region of attraction. Similarly, for $\mathcal{W} = \mathbb{R}^1$, the robot's movement is dictated by sampling a Bernoulli distribution, where the choices of moving positively or negatively from the steady-state have equal probability. Such a method can also be used to escape a region where attractors equally "pull" the robot towards them, thus the robot remains stationary.

In the next section, an overview of the Bayesian filter is presented and then the derivation of the filtering equations for the search problem given in this thesis.

2.2 The Bayesian Filter

2.2.1 Overview

As discussed in Chapter 1 the Bayesian filter is a powerful tool in search and its related applications. It stems from the widely celebrated Bayes' theorem which in part led to the development of Bayesian inference [30]. Bayesian inference determines the probability of a given hypothesis, which is referred to as the *posterior* probability, from the *prior* probability of said hypothesis

and a *likelihood function*. Explicitly, the general Bayesian inference equation, the root of which is given by the Bayes' theorem, is

$$P(B|A) = \frac{P(A|B)\, P(B)}{P(A)} \tag{2.4}$$

where $P(B|A)$ is the posterior probability distribution, $P(B)$ is the prior probability distribution, $P(A|B)$ is the likelihood function, and $P(A)$ is the marginal probability of the event A.

Given this formulation, it is possible to model the evolution of the posterior probability distribution given the above prior distribution and probability measures. Consider the case where the prior is the target distribution, i.e. B represents the object state x^o, where $\mathrm{f}_{\boldsymbol{x^o}}\,(x^o)$ is the probability distribution over x^o. Thus the posterior probability is the conditional probability that represents the new target distribution given the information represented by the event A after a single update (an update refers to computing a posterior distribution for a single time-step). Let the event A represent the binary sensor information, i.e. the sensor output given by the observations of $\boldsymbol{y_1} = 1$ or $\boldsymbol{y_1} = 0$ at time $t = 1$. Then, the probability of $\boldsymbol{y_1}$ conditioned on the object state x^o is given by the probability mass function $p_{\boldsymbol{y}_1|\boldsymbol{x^o}}(y_1|x^o)$, which is the likelihood function. Thus the Bayesian inference equation becomes

$$\mathrm{f}_{x^o|y_1}\,(x^o|y_1) = \eta_1\, p_{\boldsymbol{y}_1|\boldsymbol{x^o}}(y_1|x^o)\, \mathrm{f}_{\boldsymbol{x^o}}\,(x^o)$$

where the inverse of η_1 denotes the marginal probability, which is also known as the *normalization constant* [11]. In most cases, the marginal probability cannot be directly computed or is too complex [14]. However, the normalization constant is usually computed by utilizing the total law of probability to condition on a random variable. In this case, the probability $p(y_1)$ is conditioned on the object's position yielding

$$p(y_1) = \int p_{\boldsymbol{y}_1|\boldsymbol{x^o}}(y_1|x^o)\, \mathrm{f}_{\boldsymbol{x^o}}\,(x^o) dx^o$$

where the integral contains the likelihood function and target distribution. The limits of integration, which are given by $\mathcal{W}$, are the support of the target distribution. Specifically, there are two possible ways to compute η_1: this depends on the value of the sensor measurement obtained at time $t = 1$.

After applying the filtering equation for the first time-step, the obtained posterior is the new probability representation for the object. A new sensor measurement is taken, which intuitively affects the current hypothesis of the object location. Thus, each sensor measurement represents the time-step, and the current posterior can be updated accordingly, yielding the evolution of the object hypothesis.

Note that the simple example above does not incorporate the robot's position or the control input. To thoroughly model the search problem, presented in the thesis, this information is necessary as both the sensor model and the posterior probability are dependent on the robot's position. To include such information, it is sufficient to use the *information state* [11, 31]; this is the state consisting of the sensor history and the action/control history. Thus the canonical information state at time t is given as

$$I_t = \{x_0^r, \mathrm{f}_{\boldsymbol{x}^o}, u_0, \cdots, u_{t-1}, y_1, \cdots, y_t\} \tag{2.5}$$

where x_0^r is the initial robot position and the target distribution is included therein. However, the information state used in Section 2.2.2 is modified as

$$\tilde{I}_t = \{\mathrm{f}_{\boldsymbol{x}^o}, x_0^r, x_1^r, \cdots, x_t^r, y_1, \cdots, y_t\}$$

which is completely equivalent to the standard information state. This is due to the robot's deterministic motion model because the positions can be reconstructed from the control inputs.

The object state random variable is conditioned on the modified information state, resulting in the distribution termed as the *belief*. The belief is the PDF of the object state, which stems from the consequence that the object state is not directly measurable. Belief is another conventional term for the posterior probability. Thus, for the search problem stated, the belief is defined as

$$b_t(x^o) := \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{I}_t}(x^o|I_t) \tag{2.6}$$

This overview yields prerequisites for the evolution of the object belief. The next section explicitly gives the derivation of the belief filtering equations using the standard Bayesian development.

2.2.2 Generic Bayesian Update Law

This section derives the belief filtering equations using the standard Bayesian framework and the parameters previously discussed. As the state of the robot is completely known by the deterministic nature of its motion model, i.e. x^r is not a random variable, the belief of the object's location is given by Equation 2.6. Furthermore, the belief only represents the state of the object and it is defined, for convenience, as the *object map*

$$m_t(x^o) := \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{I}_t}\left(x^o|I_t\right) \tag{2.7}$$

which is the posterior PDF of the object location conditioned on the information vector. Though completely equivalent to Equation 2.6, this definition is given to emphasize that $m_t(x^o)$ specifically represents the object state.

Before the application of Bayes' theorem, notice that the object map can be deconstructed in the following form

$$m_t(x^o) := \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{I}_t}\left(x^o|I_t\right) = \mathrm{f}_{\boldsymbol{x}^o|\tilde{\boldsymbol{I}}_t}\left(x^o|\tilde{I}_t\right)$$

because the standard and modified information states are completely equivalent. Additionally, by manipulation of the modified information state to an equal representation of identical events

$$m_t(x^o) = \mathrm{f}_{\boldsymbol{x}^o|\boldsymbol{y}_t,x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|y_t,x_t^r,\tilde{I}_{t-1}\right)$$

Now begins the standard Bayesian approach. Applying Bayes' theorem, the object map representation is

$$m_t(x^o) = \frac{\mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x^o,x_t^r,\tilde{I}_{t-1}\right)\mathrm{f}_{\boldsymbol{x}^o|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|x_t^r,\tilde{I}_{t-1}\right)}{\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x_t^r,\tilde{I}_{t-1}\right)}$$

From the sensor model, the observation $\boldsymbol{y}_t$ is conditionally independent of $\tilde{I}_{t-1}$ when conditioned on $\boldsymbol{x}^o$ and x_t^r. Reflecting this statement by changing the likelihood function in the object map yields

$$m_t(x^o) = \frac{\mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x}^o,x_t^r}\left(y_t|x^o,x_t^r\right)\mathrm{f}_{\boldsymbol{x}^o|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|x_t^r,\tilde{I}_{t-1}\right)}{\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x_t^r,\tilde{I}_{t-1}\right)}$$

Furthermore, the robot's current position does not affect the belief of object location as long as no sensor measurement is taken. Consider the robot following some search trajectory in $\mathcal{W}$. If no observation is taken, the object map is left unchanged. However, whenever an observation is made, the probability distribution describing the object must reflect the new information observed. Thus, the prior in the object map update rule can be simplified as

$$m_t(x^o) = \frac{\mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x^o},x_t^r}\left(y_t|x^o,x_t^r\right)\mathrm{f}_{\boldsymbol{x^o}|\tilde{\boldsymbol{I}}_{t-1}}\left(x^o|\tilde{I}_{t-1}\right)}{\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\tilde{\boldsymbol{I}}_{t-1}}\left(y_t|x_t^r,\tilde{I}_{t-1}\right)}$$

The normalization constant η_t is computed by first using the total law of probability and conditioning the denominator of the above equation on x^o, then taking its inverse. Also, note that the prior is exactly the given definition of the object map in Equation 2.7, which yields

$$m_t(x^o) = \eta_t \ \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x^o},x_t^r}\left(y_t|x^o,x_t^r\right) \ m_{t-1}(x^o) \tag{2.8}$$

with the normalization constant

$$\eta_t := \left[\int \mathrm{p}_{\boldsymbol{y}_t|\boldsymbol{x^o},x_t^r}\left(y_t|x^o,x_t^r\right) \ m_{t-1}(x^o)dx^o\right]^{-1} \tag{2.9}$$

Thus, using the Bayesian inference development, a generic update law has been given for the specified search problem.

Though the model in this chapter is discussed in terms of a single searcher, it is extended towards the use of multiple agents. This extension is relatively straightforward without changing the model presented in the thesis. Such an extension stems from the intuition that multiple agents share a common belief of the object [22, 25]. As each agent progresses through $\mathcal{W}$, individually following the gradient ascent strategy, a binary sensor measurement is independently made at some time t, resulting in the set of observations

$$\boldsymbol{Y}_t = \{\boldsymbol{y}_t^1, \boldsymbol{y}_t^2, \cdots, \boldsymbol{y}_t^n\}$$

where the superscript n denotes the number of search agents. By the general Bayesian filtering update, given in Equation 2.8, each observation is separately used to generate a new posterior. This is akin to a single searcher

having the ability to "transport" to a new location before updating the object map. Much more information is thus available for the evolution of the object map, and the ascent control law dictates the next movement for each searcher. Additionally, the specific update laws and the approximation method given in Chapter 3 are valid for the use of multiple agents because they share a common object map. The effects and results of multiple agents following the gradient ascent strategy are discussed in Chapter 4 along with single agent search.

This concludes the general formulation of search and Chapter 2. The thesis continues by presenting the exponential family class of probability distributions and its property of self-conjugacy. Also, the sensor model (likelihood) specifics are outlined. Using the general formulation given above, an exact parameterization that encompasses the class of exponential families of the object PDF is derived. This parameterization is further demonstrated employing mixtures of Gaussians. Finally, an approximation method for the object map is outlined and given at the end of the chapter.

3 BELIEF REPRESENTATION

This chapter introduces the exponential family class of probability distributions. An overview of their self-conjugacy property is also presented. The details concerning the sensor model are also given, and a closed-form update law is proven for the previously derived Bayesian filtering law. This law is derived generally using the self-conjugacy property of exponential families for an atypical mixture model. Additionally, this law is demonstrated utilizing a mixture of Gaussians. Finally, due to the possible exponential parameter growth of its closed-form parameterization, the evolution of the object map is approximated by employing particle filtering.

3.1 Exponential Families and Self-Conjugacy

The exponential family of distributions is a class of probability distributions that can all be represented in a single form. This form encompasses many known continuous and discrete distributions in either their univariate or multivariate parameterizations (dependent on the sample space). Exponential families are also known as Koopman-Darmois families based on the pioneering work in [32–34]. Some well-known distributions that are members of this class include the Gaussian, Rayleigh, exponential, binomial, Bernoulli, beta, gamma, and Poisson distributions.

The canonical parameterization, in the form of a PDF, is given as

$$\mathrm{f}\left(x;\theta\right) = \exp\left[\langle\theta, T(x)\rangle - F(\theta)\right] \tag{3.1}$$

where the brackets $\langle\ \rangle$ indicate the dot product operation, θ is a vector containing the *natural parameters*, $T(x)$ is the *sufficient statistic*, and $F(\theta)$ is the *log normalizer* [35]. The sufficient statistic is a reduced form of the sample set which provides all necessary information to recover a distribution's parameters. That is, when the sufficient statistic is known, knowledge of any

other statistic from the sample set provides no new information about the distribution [32,34]. For exponential families, it is possible to represent $T(x)$ as a scalar (or vector) where its dimensionality is equal to the dimensionality of the natural parameters. For a given distribution within the exponential family, the natural parameters describe all of the parameters needed to represent said distribution. The log normalizer parameter characterizes the normalization constant for the exponential family form. Furthermore, this parameter is given by a function of the natural parameters, where this function is differentiable and strictly convex.

The parameterization is illustrated with the well-known exponential distribution [27] with its PDF given as

$$\mathrm{f}\left(x;\lambda\right) = \lambda \exp\left[-\lambda x\right]$$

Given the sufficient statistic vector as

$$T(x) = [x, 0]$$

the natural parameter vector is simply

$$\theta = [\theta_1, \theta_2] = [-\lambda, 0]$$

The form of the log normalizer parameter is derived by first representing the exponential PDF explicitly in exponential family form

$$\begin{aligned}\mathrm{f}\left(x;\lambda\right) &= \lambda \exp\left[-\lambda x\right] \\ &= \lambda \exp\left[\langle \theta, T(x)\rangle\right]\end{aligned}$$

where it is desired to "move" the λ coefficient inside the exponential. This can be accomplished by the following

$$\begin{aligned}\mathrm{f}\left(x;\lambda\right) &= \exp\left[\ln(\lambda)\right] \cdot \exp\left[\langle \theta, T(x)\rangle\right] \\ &= \exp\left[\langle \theta, T(x)\rangle + \ln(\lambda)\right] \\ &= \exp\left[\langle \theta, T(x)\rangle - (-\ln(\lambda))\right] \\ &= \exp\left[\langle \theta, T(x)\rangle - F(\theta)\right]\end{aligned}$$

where the log normalizer function is simply

$$F(\theta) = -\ln(-\theta_1)$$

to ensure that the log normalizer, as a function of θ, is convex and differentiable since $\lambda > 0$. Thus the exponential PDF has been parameterized in canonical exponential family form.

An important, relevant property of the exponential family is its *self-conjugacy* under the operation of the Bayesian filter. That is, given a prior, belonging to the exponential family class, it is called a *conjugate prior* if the posterior distribution can be parameterized in the same form [35, 36]. This property occurs for exponential families when the likelihood is also in exponential family form. Furthermore, it is explicitly shown in Section 3.3.1 that the posterior distribution can be parameterized as a mixture in canonical exponential family form given both an exponential family prior distribution and likelihood. This is done assuming the prior is a special case of an exponential family mixture distribution rather than a single distribution.

This concludes the overview of the class of exponential family distributions. The next section describes the sensor model specifics necessary for the belief representation given in the following sections.

3.2 Likelihood of Detection

In this section the mobile robot's sensing model is defined. As discussed previously, the sensor model yields the observation $\boldsymbol{y}_t$, where $\boldsymbol{y}_t = \{1, 0\}$ for object detection or absence, respectively, at time t. The calibrated sensor model for detection conditioned on the robot's current position and object location is parameterized in the canonical exponential family form as

$$\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 1|x_t^r, x^o\right) = \kappa \exp\left[\langle\theta_s, T(x^o)\rangle - F(\theta_s)\right] \tag{3.2}$$

where κ is chosen such that the maximum value of this function is less than or equal to one, θ_s is a vector of the natural parameters, $T(x^o)$ is the sufficient statistic vector, and $F(\theta_s)$ is the log normalizer. Though intuitive, note that

the observation of $\boldsymbol{y}_t = 0$ yields the following form

$$\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 0|x_t^r, x^o\right) = 1 - \kappa \exp\left[\langle \theta_s, T(x^o)\rangle - F(\theta_s)\right] \tag{3.3}$$

which is simply the complement of the probability of object detection given in Equation 3.2. Section 3.3.2 presents the specific case of an initial prior composed of a mixture of Gaussians with the corresponding sensor model given as

$$\mathrm{p}_{\boldsymbol{y}_t|x_t^r,\boldsymbol{x}^o}\left(\boldsymbol{y}_t = 1|x_t^r, x^o\right) = \mathcal{N}(x_t^r, \Sigma_s) \tag{3.4}$$

where the Gauassian distribution is univariate for $\mathcal{W} = \mathbb{R}^1$ and multivariate for $\mathcal{W} = \mathbb{R}^2$. The exponential family parameterization [37] of the univariate Gaussian distribution is expressed as

$$\begin{aligned}
\theta_s &= \left[\frac{x_t^r}{\sigma_s^2}, -\frac{1}{2\sigma_s^2}\right] \\
F(\theta_s) &= \frac{(x_t^r)^2}{2\sigma_s^2} + \frac{1}{2}\ln(2\pi\sigma_s^2) \\
T(x^o) &= \left[x^o, (x^o)^2\right]^T
\end{aligned}$$

and multivariate parameterization is expressed as

$$\begin{aligned}
\theta_s &= \left[\Sigma_s^{-1}x_t^r, \frac{1}{2}\Sigma_s^{-1}\right] \\
F(\theta_s) &= \frac{1}{4}\mathrm{Trace}\left(\left(\frac{1}{2}\Sigma_s^{-1}\right)^{-1}\Sigma_s^{-1}x_t^r\left(\Sigma_s^{-1}x_t^r\right)^T\right) + \ln(\pi) \\
&\quad - \frac{1}{2}\ln\left(\left|\frac{1}{2}\Sigma_s^{-1}\right|\right) \\
T(x^o) &= \left[x^o, -x^o(x^o)^T\right]^T
\end{aligned}$$

Here note the importance of κ for the univariate case: for the previously given sensor model in Equation 3.4, if

$$\sigma_s < \frac{1}{\sqrt{2\pi}}$$

then the sensor model is not a valid likelihood because it does not integrate to one over $\mathcal{W}$. Thus κ is used to enforce this necessary condition.

The next section demonstrates that weighted sums of distributions from an

exponential family are self-conjugate by proof and parameterizes the object map update law.

3.3 Belief Update Laws

3.3.1 General Exponential Families

This section explicitly shows that, under certain conditions, the generic Bayesian update law allows the parameterization of the object map in the exponential family class of distributions. This parameterization is exact and finite for a finite number of stages. Although this result is explicitly demonstrated, it can also be inferred from the form of Equation 2.8 and the self-conjugacy property of distributions for the exponential family under the Bayesian filter. However, this proof shows that the self-conjugacy property of a mixture of exponential families further holds when the mixture component weights are not given in standard form, i.e. not strictly greater than zero.

Theorem 1. *A non-standard prior and likelihood belonging to the exponential family class are sufficient conditions for a conjugate prior of the object map given by Equation 2.8. Thus the posterior will be a finite parameterization of the same distribution class as the prior.*

Proof. First, the initial prior is given as a standard mixture of exponential family distributions to represent the probable object locations over $\mathcal{W}$, given as the convex combination

$$m_0(x^o) = \sum_{i=1}^{n} w_i \exp\left[\langle \theta_i, T(x^o) \rangle - F(\theta_i)\right] \tag{3.5}$$

where w_i is the i^{th} mixture component weight satisfying

$$\sum_{i}^{n} w_i = 1, \; w_i > 0, \quad \forall i$$

Note that, in their standard form, the mixture weights w_i are constrained to the set of positive reals. Previous work utilizes an initial prior (target distribution) as a mixture of Gaussians, which can be represented by Equa-

tion 3.5 because the Gaussian distribution belongs to the exponential family class [2, 21, 22, 24].

However, the proof demonstrates the theorem inductively by defining an atypical mixture [38], where each component's weight is not constrained to the set of positive reals. Thus, an atypical mixture of exponential family distributions is given as

$$m_t(x^o) := \sum_{i=1}^{n} v_i \exp\left[\langle \theta_i, T(x^o) \rangle - F(\theta_i)\right] \tag{3.6}$$

where v_i is a component weight of the atypical mixture. Additionally, these weights must fulfill the following requirements

$$\sum_{i}^{n} v_i = 1, \ v_i \in \mathbb{R}, \quad \forall i$$

and

$$m_t(x^o) \geq 0, \quad \forall x^o \in \mathcal{W}$$

to ensure that the mixture is a valid probability density function. Essentially, this is a mixture of "positive" components summed with "negative components" (see Figure 3.1). Hence, the weighting coefficients of the mixture components are denoted with v_i to emphasize that this mixture is not the usual convex sum of component distributions. Furthermore, notice that any distribution of the form of Equation 3.5 can be written in the form of Equation 3.6 without modification, so the initial condition of the induction proof is satisfied given the exponential family mixture with positive components. The proof continues by demonstrating the inductive step first for the case of $\boldsymbol{y}_t = 1$. Substituting the sensor model (Equation 3.2) into the generic Bayesian update law (Equation 2.8) yields

$$m_t(x^o) = \eta_t \ \kappa \exp\left[\langle \theta_s, T(x^o) \rangle - F(\theta_s)\right] \ m_{t-1}(x^o)$$

Using the inductive hypothesis that the object map $m_{t-1}(x^o)$ is a mixture of

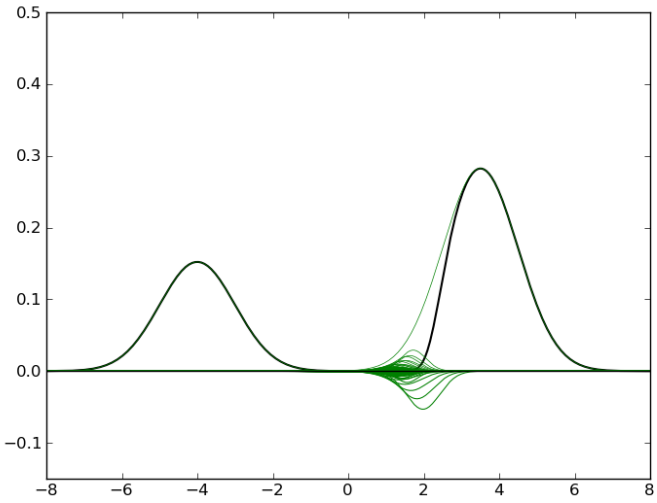


Figure 3.1: Example of the atypical mixture model. The black line represents the sum of the mixture components. The green lines depict each component separately.

exponential family distributions gives

$$m_t(x^o) = \eta_t \ \kappa e^{\langle \theta_s, T(x^o) \rangle - F(\theta_s)} \ \sum_{i=1}^{n} \upsilon_i e^{\langle \theta_i, T(x^o) \rangle - F(\theta_i)}$$
$$= \eta_t \ \kappa \ \sum_{i=1}^{n} \upsilon_i e^{\langle \theta_i + \theta_s, T(x^o) \rangle - F(\theta_i) - F(\theta_s)}$$

which is accomplished by manipulating the sum and using the exponentiation identity of the exponential function. However, this form is not yet parameterized in the canonical exponential family representation due to the combination of log normalizer parameters. Thus, α_i is defined as

$$\alpha_i := F(\theta_i + \theta_s) - F(\theta_i) - F(\theta_s)$$

which is simply the difference of the computed log normalizer $F(\theta_i + \theta_s)$ and the log normalizers of the i^{th} component and sensor model. Now the mixture model can be rewritten as

$$m_t(x^o) = \eta_t \ \sum_{i=1}^{n} \kappa \upsilon_i e^{\alpha_i} \ e^{\langle \theta_i + \theta_s, T(x^o) \rangle - F(\theta_i + \theta_s)}$$

Rather than computing η_t with Equation 2.9, it is computed based on its normalizing property alone, i.e.

$$\begin{aligned}\eta_t &= \left[\int \sum_{i=1}^{n} \kappa v_i e^{\alpha_i}\ e^{\langle \theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)} dx^o\right]^{-1} \\ &= \left[\sum_{i=1}^{n} \kappa v_i e^{\alpha_i}\ \int e^{\langle \theta_i+\theta_s, T(x^o)\rangle - F(\theta_i+\theta_s)} dx^o\right]^{-1} \\ &= \left[\sum_{i=1}^{n} \kappa v_i e^{\alpha_i}\right]^{-1} \end{aligned} \tag{3.7}$$

The final step is performed by noting each component is a probability distribution: they each integrate to one. Thus, the object map at time t can be parameterized as

$$m_t(x^o) = \sum_{i=1}^{n} v_i^+ \exp\left[\langle \theta_i^+, T(x^o)\rangle - F(\theta_i^+)\right] \tag{3.8}$$

where the atypical weighting coefficient v_i^+ is

$$v_i^+ = \eta_t \kappa v_i e^{\alpha_i}$$

The superscript "+" indicates the component in the stage t object map with parameters (v_i^+, θ_i^+) is generated (through the Bayesian filtering equation) by the component in the stage $t-1$ object map with parameters (v_i, θ_i) given an observation of $\boldsymbol{y}_t = 1$. This satisfies the first case of the proof as Equation 3.8 is in canonical exponential family form.

The second case of $\boldsymbol{y}_t = 0$ is considered with the same inductive reasoning presented above. Substituting the sensor model complement (Equation 3.3) into the generic Bayesian update law (Equation 2.8) yields

$$m_t(x^o) = \eta_t \left(1 - \kappa e^{\langle \theta_s, T(x^o)\rangle - F(\theta_s)}\right) m_{t-1}(x^o)$$

Utilizing the inductive hypothesis and using the same steps as the previous

case,

$$
\begin{aligned}
m_t(x^o) &= \eta_t m_{t-1}(x^o) - \eta_t \sum_{i=1}^{n} \kappa v_i e^{\alpha_i} e^{\langle \theta_i + \theta_s, T(x^o) \rangle - F(\theta_i + \theta_s)} \\
&= \sum_{i=1}^{n} v_i^+ e^{\langle \theta_i^+, T(x^o) \rangle - F(\theta_i^+)} + \sum_{i=1}^{n} v_i^- e^{\langle \theta_i^-, T(x^o) \rangle - F(\theta_i^-)} \qquad (3.9)
\end{aligned}
$$

where η_t can be computed by simply taking the complement of Equation 3.7, i.e. subtract by one. The superscript "+" retains the meaning as before and similarly the superscript "−" indicates a component with a negative weighting coefficient. For the case of either observation, the property $m_t(x^o) \geq 0$ holds as a consequence of Bayes' theorem. Thus, for the second observation case, the induction step holds and is able to represent m_t with a finite number of parameters in exponential family form. □

The proof results in intuitive update rules for each observation case. These rules are presented in the following corollaries.

Corollary 1. *For an observation of $y_t = 1$ and given the parameters (v_i, θ_i) for $m_{t-1}(x^o)$ and θ_s, the parameters for $m_t(x^o)$ are given as*

$$v_i^+ = \eta_t \ \kappa v_i e^{\alpha_i} \qquad (3.10)$$

$$\theta_i^+ = \theta_i + \theta_s \qquad (3.11)$$

The interpretation of Corollary 1 is that observations of $y_t = 1$ result in support of the previous object map that is close to the robot position being scaled up, meaning that the sensor reading is likely if the object is close. The support farther from the robot will thus be scaled down.

Corollary 2. *For an observation of $y_t = 0$ and given the parameters (v_i, θ_i) for $m_{t-1}(x^o)$ and θ_s, the parameters for $m_t(x^o)$ are given as*

$$v_i^+ = \eta_t v_i \qquad (3.12)$$

$$\theta_i^+ = \theta_i \qquad (3.13)$$

$$v_i^- = -\eta_t \ \kappa v_i e^{\alpha_i} \qquad (3.14)$$

$$\theta_i^- = \theta_i + \theta_s \qquad (3.15)$$

Essentially, in this second case, the number of components in the mixture

will double. Each component of the original mixture will produce a scaled copy of itself (represented by parameters v_i^+, θ_i^+) and a component whose coefficient has opposite sign (represented by parameters v_i^- and θ_i^-). This behavior can be seen again in Figure 3.1.

Both corollaries yield intuitive update laws for implementation. These are formed as a general algorithm presented in Algorithm 1 for a single time-step. In implementation it is sufficient to separate positive and negative components for bookkeeping because the object map at each time-step may contain both types of components.

Algorithm 1 Map Update for a Single Time Stage

Require: $y_t, m_{t-1}, \theta_s, \kappa$
  Compute n_t, according to Equation 3.7
  if $y_t = 1$ **then**
    for each $(v_i, \theta_i) \in m_{t-1}$ **do**
      $\theta_i^+ = \theta_i + \theta_s$
      $\alpha_i = F(\theta_i + \theta_s) - F(\theta_i) - F(\theta_s)$
      Compute v_i^+, according to Equation 3.10
      $(v_i^+, \theta_i^+) \Rightarrow m_t$
    end for
  else
    $n_t = 1 - n_t$ (*take the complement*)
    for each $(v_i, \theta_i) \in m_{t-1}$ **do**
      $\theta_i^+ = \theta_i$
      Compute v_i^+, according to Equation 3.12
      $\theta_i^- = \theta_i + \theta_s$
      $\alpha_i = F(\theta_i + \theta_s) - F(\theta_i) - F(\theta_s)$
      Compute v_i^-, according to Equation 3.14
      $\left(v_i^+, \theta_i^+, v_i^-, \theta_i^-\right) \Rightarrow m_t$
    end for
  end if
  return m_t

The exact representation, given by the update rules, produces a direct means of calculation for the control law presented in Section 2.1. Expanding

the general gradient ascent with the object map representation yields

$$\begin{aligned}\tilde{x}^r_{t+1} &= x^r_t + \gamma \nabla m(x^o) \\ &= x^r_t + \gamma \nabla \left(\sum_{i=1}^{n} v_i^+ e^{\langle \theta_i^+, T(x^o) \rangle - F(\theta_i^+)} + \sum_{i=1}^{n} v_i^- e^{\langle \theta_i^-, T(x^o) \rangle - F(\theta_i^-)} \right) \\ &= x^r_t + \gamma \sum_{i=1}^{n} v_i^+ \nabla \left(e^{\langle \theta_i^+, T(x^o) \rangle - F(\theta_i^+)} \right) + \sum_{i=1}^{n} v_i^- \nabla \left(e^{\langle \theta_i^-, T(x^o) \rangle - F(\theta_i^-)} \right)\end{aligned} \tag{3.16}$$

This holds by the linearity of the gradient operator and the assumption that the object map at time t consists of both positive and negative components. Then the control input can be chosen accordingly, given the method in Section 2.1. For the implementation of an initial belief comprised of Gaussian components in Section 3.3.2, the gradient ascent for the canonical exponential family form still holds. Additionally, the gradient of an atypical mixture of Gaussians is explicitly given. Furthermore, as demonstrated in Section 3.4, the approximation of the object map yields a mixture of Gaussians, thus allowing the continued usage of the gradient calculation and ascent strategy.

Though it has been demonstrated that the object map can be represented with a finite representation, its size is not fixed. Considering the update rule given $\boldsymbol{y}_t = 1$, the number of components will not change. However, notice for the case of $\boldsymbol{y}_t = 0$ that the number of components will double. Specifically, the worst-case scenario for component growth is comprised of a series of successive negative detections, i.e. the robot is far away from the object. Thus the number of components that may be required is $n2^t$ at stage t, where n is the number of components present in the initial distribution ($m_0(x^o)$). This exponential growth of the representation may cause serious computational issues in practice if the exact object map representation is solely used. To reduce this representation's complexity, Section 3.4 describes the approximation method chosen, namely, regularized particle filtering.

This concludes the belief representation derivations that utilize the exponential family parameterization. The next section presents the evolution of the exact belief using an atypical mixture of Gaussian distributions.

3.3.2 Mixture of Gaussians

In this section the belief update rules are demonstrated explicitly using the exponential family representation for an atypical mixture of Gaussian distributions over $\mathcal{W} = \mathbb{R}^2$. For a multi-variate Gaussian distribution with mean vector μ and covariance matrix Σ, the natural parameters θ, and sufficient statistic $T(x^o)$, the exponential family representation is

$$\begin{aligned} \theta = [\theta_1, \theta_2] &= \left[\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1}\right] \\ T(x^o) &= \left[x^o, -x^o(x^o)^T\right]^T \end{aligned} \tag{3.17}$$

Here the notation for θ_1 and θ_2 does not denote the specific mixture component. Rather, the subscripts represent the row vector representation of θ. The log normalizer is expressed in terms of the natural parameters of the distribution.

$$F = \frac{1}{4}\text{Trace}\left[\left(\frac{1}{2}\Sigma^{-1}\right)^{-1}\left(\Sigma^{-1}\mu\right)\left(\Sigma^{-1}\mu\right)^T\right] - \frac{1}{2}\ln\left[\det\left(\frac{1}{2}\Sigma^{-1}\right)\right] + \frac{1}{2}\ln(\pi)$$

Once the initial distribution is "converted" from a mixture of Gaussians to the canonical representation in exponential family form, Equations 3.8 or 3.9 can be used to model the belief evolution given the sensor observation.

Of course the mixture of exponentials representation can be converted back to a mixture of Gaussians using the mapping from natural parameters to mean and covariance.

$$\mu = \frac{1}{2}\theta_1^{-1}\theta_2 \tag{3.18}$$

$$\Sigma = \frac{1}{2}\theta_2^{-1} \tag{3.19}$$

Using this mapping, the update rule for $y_t = 1$ for a single time-step is

$$v_i^+ = \eta_t \ \kappa v_i e^{-\alpha_i}$$

as the weights are unchanged from Equation 3.10. The mean μ_i^+ and covari-

ance Σ_i^+ of each component are written as

$$\mu_i^+ = \frac{1}{4}(\Sigma_i^{-1}\mu_i + \Sigma_s^{-1}\mu_s)^{-1}(\Sigma_i^{-1} + \Sigma_s^{-1})$$
$$\Sigma_i^+ = (\Sigma_s^{-1} + \Sigma_i^{-1})^{-1}$$

by substituting Equations 3.17 to 3.19 into Equation 3.11. Similarly, the update rule for $y_t = 0$ for a single time-step can be written as

$$\begin{aligned} v_i^+ &= \eta_t v_i & v_i^- &= -\eta_t\ \kappa v_i e^{-\alpha_i} \\ \mu_i^+ &= \mu_i & \mu_i^- &= \frac{1}{4}(\Sigma_i^{-1}\mu_i + \Sigma_s^{-1}\mu_s)^{-1}(\Sigma_i^{-1} + \Sigma_s^{-1}) \\ \Sigma_i^+ &= \Sigma_i & \Sigma_i^- &= (\Sigma_s^{-1} + \Sigma_i^{-1})^{-1} \end{aligned}$$

This can be shown by substituting Equations 3.17 to 3.19 into Equations 3.12 to 3.15. It is also interesting to note that update rules for the covariance matrices are similar to the Kalman filter covariance update [11]. Such a similarity is not surprising as, discussed in the introduction, the Kalman filter incorporates the Bayesian filter in the update step.

Once the atypical mixture of Gaussians is obtained from the modified update rules above,

$$m_t(x^o) = \sum_{i=1}^{n} v_i^+ G(x^r; u_i^+, \Sigma_i^+) + \sum_{i=1}^{n} v_i^- G(x^r; u_i^-, \Sigma_i^-) \tag{3.20}$$

where the function G is given in Equation 3.21, the gradient ascent control can be explicitly defined. This is an extension of the general exponential family gradient, given in Equation 3.16, to the atypical mixture of Gaussians.

Before explicitly computing the gradients of the atypical mixture, the function $G(x; \mu, \Sigma)$ is presented for clarity, where x is the argument of the function given the parameters μ and σ. This function is simply the well-known parameterization of the bivariate Gaussian

$$G(x; \mu, \Sigma) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} \exp\left(\frac{-(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}\right) \tag{3.21}$$

for $\mathcal{W} = \mathbb{R}^2$, where x is a two-dimensional column vector denoted as $[x_1, x_2]^T$. The mean μ is also a two-dimensional column vector and the symmetric

covariance matrix is of the form

$$\Sigma = \begin{bmatrix} \sigma_{x_1} & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2} \end{bmatrix}$$

From Equation 3.20 the mixture of Gaussians is substituted into Equation 3.16

$$\tilde{x}^r_{t+1} = x^r_t + \gamma \left(\sum_{i=1}^{n} v_i^+ \nabla \left(G(x^r; \mu_i^+, \Sigma_i^+) \right) + \sum_{i=1}^{n} v_i^- \nabla \left(G(x^r; \mu_i^-, \Sigma_i^-) \right) \right) \tag{3.22}$$

which again holds by the linearity of the gradient operator. The gradient of the function G is given as a two-dimensional column vector

$$\nabla \left(G(x^r; \mu, \Sigma) \right) = \left[\frac{\partial G}{\partial x^r_1}, \frac{\partial G}{\partial x^r_2} \right]^T$$

where it is evaluated at the current position of the robot at time t (the time subscript has been temporarily omitted to avoid confusion in the forthcoming equations). Thus, the gradient of the object map, as an atypical mixture of Gaussians, is the sum of partial derivatives

$$\nabla m_t(x^o) = \sum_{i=1}^{n} v_i^+ \left[\frac{\partial G}{\partial x^r_1}, \frac{\partial G}{\partial x^r_2} \right]^T + \sum_{i=1}^{n} v_i^- \left[\frac{\partial G}{\partial x^r_1}, \frac{\partial G}{\partial x^r_2} \right]^T \tag{3.23}$$

Here the positive and negative component partials are given separately. For the sum of positive components, the gradient yields the column vector

$$\sum_{i=1}^{n} v_i^+ \left[\frac{\partial G}{\partial x^r_1}, \frac{\partial G}{\partial x^r_2} \right]^T =$$

$$- \sum_{i=1}^{n} \frac{v_i^+ G(x^r; \mu_i^+, \Sigma_i^+)}{\sigma^+_{i,x_1} \sigma^+_{i,x_2} - (\sigma^+_{i,x_1 x_2})^2} \cdot \begin{bmatrix} \sigma^+_{i,x_2}(x^r_1 - \mu^+_{i,1}) - \sigma^+_{i,x_1 x_2}(x^r_2 - \mu^+_{i,2}) \\ \\ \sigma^+_{i,x_1}(x^r_2 - \mu^+_{i,2}) - \sigma^+_{i,x_1 x_2}(x^r_1 - \mu^+_{i,1}) \end{bmatrix}$$

and likewise the gradient of the sum of negative components yields

$$\sum_{i=1}^{n} v_i^- \left[\frac{\partial G}{\partial x_1^r}, \frac{\partial G}{\partial x_2^r}\right]^T =$$

$$-\sum_{i=1}^{n} \frac{v_i^- G(x^r; \mu_i^-, \Sigma_i^-)}{\sigma_{i,x_1}^- \sigma_{i,x_2}^- - (\sigma_{i,x_1x_2}^-)^2} \cdot \begin{bmatrix} \sigma_{i,x_2}^-(x_1^r - \mu_{i,1}^-) - \sigma_{i,x_1x_2}^-(x_2^r - \mu_{i,2}^-) \\ \\ \sigma_{i,x_1}^-(x_2^r - \mu_{i,2}^-) - \sigma_{i,x_1x_2}^-(x_1^r - \mu_{i,1}^-) \end{bmatrix}$$

Furthermore, the partials of the components take a more direct form under certain conditions, particularly arising from the structure of the object map and sensor model covariance matrices. One such condition is that the object map is comprised of circular Gaussian components, each with a covariance matrix of the form

$$\Sigma_i = \sigma_i^2 \cdot I \tag{3.24}$$

where σ_i^2 is the variance and I denotes the two-dimensional identity matrix. Moreover, the sensor model covariance matrix is given as

$$\Sigma_s = \sigma_s^2 \cdot I \tag{3.25}$$

From the positive and negative covariance update laws, it is obvious that a prior belief of circular Gaussian components and the above sensor model covariance matrix yield a posterior where the component covariance matrices remain in the general form given in Equation 3.24. The forthcoming equations use this form to clearly describe the computed gradients of each component, though the scaling of the component covariance matrix may change as the object map evolves. Therefore, the positive component partials are of the form

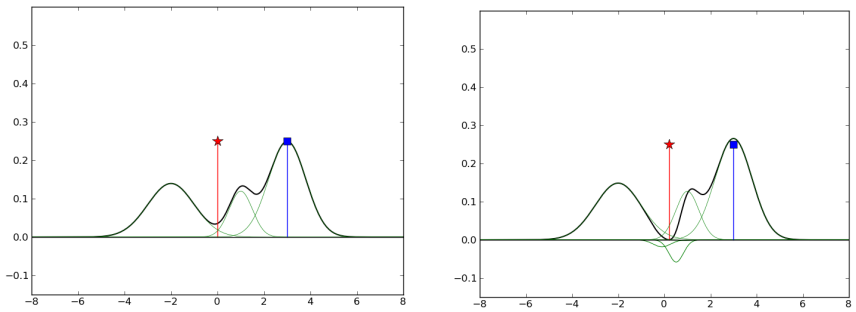
$$\sum_{i=1}^{n} v_i^+ \left[\frac{\partial G}{\partial x_1^r}, \frac{\partial G}{\partial x_2^r}\right]^T = -\sum_{i=1}^{n} \frac{v_i^+ G(x^r; \mu_i^+, \Sigma_i^+)}{(\sigma_i^+)^2} \cdot \left[(x_1^r - \mu_{i,1}^+), (x_2^r - \mu_{i,2}^+)\right]^T$$

and additionally the partials of the negative components are

$$\sum_{i=1}^{n} v_i^- \left[\frac{\partial G}{\partial x_1^r}, \frac{\partial G}{\partial x_2^r}\right]^T = -\sum_{i=1}^{n} \frac{v_i^- G(x^r; \mu_i^-, \Sigma_i^-)}{(\sigma_i^-)^2} \cdot \left[(x_1^r - \mu_{i,1}^-), (x_2^r - \mu_{i,2}^-)\right]^T$$

Thus, the computed component gradients above can be substituted into Equation 3.22 to complete the ascent control law.

The behavior of this control law can be discussed simply in terms of each component given the previous derivations. The gradients of the positive components act as expected, which "pulls" the robot towards each component's respective mean. The magnitude of attraction is, intuitively, directly proportional to the weights and the object likelihood at the evaluated position but inversely proportional to the variance of each component. As the variance of a component increases, it decreases the attractive effect of said component. Remembering that the negative components are negatively weighted, these components have a repulsive effect upon the robot, thus producing the opposite effect than their positive counterparts. That is, a region where negative components are populated will "push" the robot from this area.



(a) $t = 0$ (b) $t = 1, y_1 = 0$

(c) $t = 4, y_4 = 0$ (d) $t = 10, y_1 0 = 0$

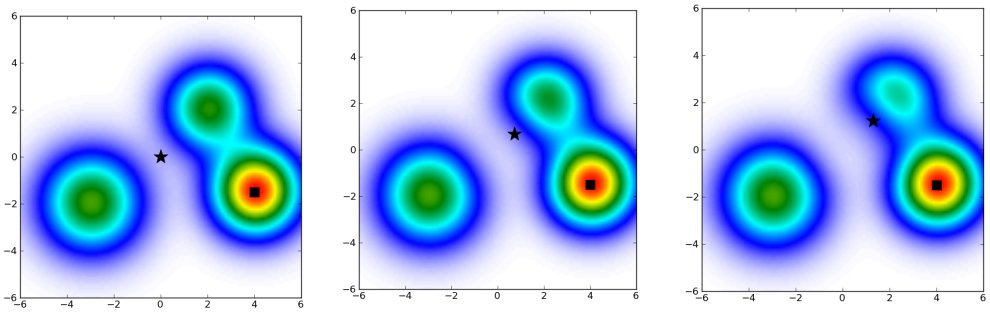
Figure 3.2: Evolution of m_t for a one-dimensional setting.

The qualitative behavior of modeling the search problem with an atypical mixture of Gaussians is given for two example systems over different workspaces. Figure 3.2 shows the time evolution of a one-dimensional map with an initial distribution based on a mixture of three Gaussian distributions.

In these figures, the thick black curve indicates the function $m_t(x^o)$ and

the thin green curves are the individual components comprising the mixture model. The impulse function marked with a red star indicates the robot's position, and the impulse function marked with a blue square indicates the object position (unknown to the robot). As time evolves, the robot moves towards the right, and, due to sensor measurements, the likelihood that the object is at the locations near sensing locations decreases. The observations governing this sample path system evolution are sampled from the sensor model conditioned on the object being located at the marked point as given in Section 3.2.

Consider Figure 3.3, which shows the time evolution of a two-dimensional map. In this case, m_t is represented by a heat map where (relatively) higher likelihood is denoted by the red areas. Again, the star marks the position of



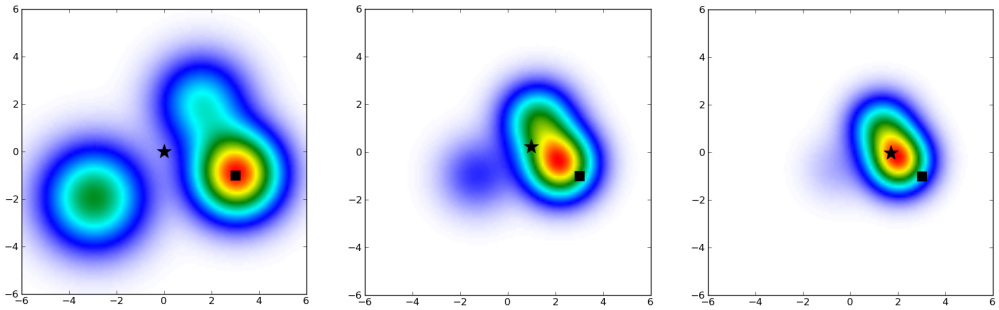
(a) $t = 0$ (b) $t = 5, y_5 = 0$ (c) $t = 9, y_9 = 0$

Figure 3.3: Evolution of m_t for a two-dimensional problem ($\mathcal{W} = \mathbb{R}^2$). Note the shrinking effect of the hypothesis given successive negative observations.

the robot and the square marks the position of the object. As time evolves, the robot follows the gradient of the map and begins "eliminating" the hypothesis that the object belongs to the Gaussian distribution in the top right of the plots, whereas all other belief is (slightly) scaled up.

Here the effects of the sensor variance are evaluated. The figures discussed previously illustrate a "tight" sensor variance, where σ_s is a small value in Equation 3.25. A positive detection is less probable the farther it is from the object and highly probable as the distance decreases. Hence, the sensor observations will mainly consist of negative detections (if the robot starts the search process far away). Again, multiple successive negative detections lead to computational issues due to the mixture model component's growth.

Inversely, a large sensor variance yields a model where the sensor measurements will be dominated by positive detections. This is demonstrated in

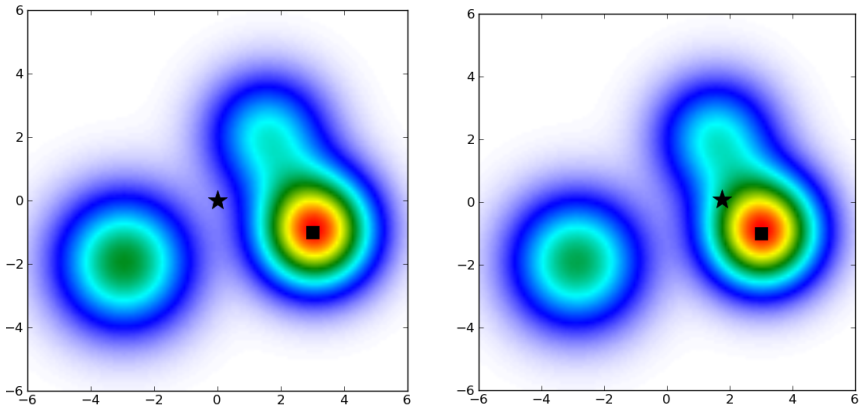


(a) $t = 0$ (b) $t = 5, y_5 = 1$ (c) $t = 9, y_9 = 1$

Figure 3.4: Evolution of m_t with a large sensor variance.

Figure 3.4 as the hypotheses closer to the robot are scaled up as the robot moves through $\mathcal{W}$, employing the gradient ascent strategy.

If the sensor variance is increased dramatically, the belief evolution does not change, as illustrated in Figure 3.5. Intuitively this makes sense if the variance is increased. For the sensor model, the probability under the tails increases (farther from the mean) as well. This again leads to sensor observations of only positive object detections throughout $\mathcal{W}$. Thus, a sensor model with a large variance does not yield new information beyond the initial belief and models a "bad" sensor. Furthermore, such a sensor model is analyzed explicitly. For the workspace of $\mathcal{W} = \mathbb{R}^1$, consider a large sensor variance,



(a) $t = 0$ (b) $t = 9, y_9 = 1$

Figure 3.5: The effect of an extremely large sensor variance on belief evolution.

specifically the case as the sensor variance approaches positive infinity

$$\theta_s = \left[\frac{\mu_s}{\sigma_s^2},\ \frac{-1}{2\sigma_s^2}\right]$$
$$\lim_{\sigma_s \to +\infty} \theta_s = \lim_{\sigma_s \to +\infty} \left[\frac{\mu_s}{\sigma_s^2},\ \frac{-1}{2\sigma_s^2}\right]$$
$$\lim_{\sigma_s \to +\infty} \theta_s = [0,\ 0]$$

Thus, by the update rules presented in Corollary 1, it is clear that the prevalent positive detection sensor model contributes no new information to the evolution of the object map. Conversely, a negative observation essentially has the opposite effect. Probability is subtracted near the robot, and the original object map is rescaled so that areas far from the robot become more likely.

This concludes the section about the belief update laws using atypical mixtures of exponential family components and Gaussian components. The next section pertains to approximating the exact representation of the belief.

3.4 Belief Approximation with Particle Filtering

Though an exact representation of the object map is available, the looming possibility of successive negative observations must be approached. As illustrated before, such an observation sequence catalyzes the exponential growth of mixture components, which is not feasible for planning a search strategy. To overcome this hurdle, approximation methods must be used in order to reduce the complexity of the object map evolution.

Particle filters, known as sequential Monte Carlo methods, approximate the posterior distribution with a finite number of parameters. These filters are not new to roboticists [11] and are an easily implementable extension of the general Bayesian filter. Generally, this approximation starts by sampling the importance function [39], where an importance function is used when sampling the distribution to approximate is intractable. For this problem, the distribution is either the prior or the object map (at some time t that the approximation is desired), so the importance function is not necessary. From this distribution, N samples (particles) are drawn and propagated through

the filtering process, which results in a particle set of the same size, i.e.

$$X_t = \{x_i, w_i\}_{i=1}^{N}$$

where X is the set of particles at time t and where the weights satisfy

$$\sum_{i=1}^{N} w_i = 1$$

Given the set X_t, the posterior distribution, using the notation for the object map, can be represented as

$$m(x^o) = \sum_{i=1}^{N} w_i \, \delta(x^o - x_i) \tag{3.26}$$

where $\delta(x^o - x_i)$ is the Dirac delta function. Thus, this yields an approximation that is a discrete random measure for the posterior distribution. Initially the weights for each particle are intuitively assigned because each particle is sampled from a mixture model, which in turn is sampled from a weighted component [37]. The initial weight associated with the particle is simply its component weight.

Out of the many flavors of particle filters the *sampling importance resampling* (SIR) filter is readily implementable [16, 40] towards the general Bayesian formulation given in Chapter 2. Again, rather than sampling from an importance density, the exact prior or object map at time t is available for the given formulation. After drawing N weighted samples from this distribution, the weights of each particle are updated with the previously defined likelihood function in Section 3.2, depending on the observation y_t. Each weight represents the *importance factor* of each particle. Then the resampling step occurs by drawing a particle, with replacement, from X_t, with the probability given by the importance factor. By drawing the particles as such, the "larger" particles (i.e. higher weights) are replicated more often than the "smaller" particles. Thus, the particles with the smallest weights are not drawn at all. After resampling a new particle set, X_{t+1} is obtained and is the same size as X_t. Finally, the weights in X_{t+1} are re-normalized, thus yielding a discrete approximation given in Equation 3.26. This process is outlined in Algorithm 2.

Algorithm 2 SIR Particle filter

Require: X_t, x_t^r, y_t

Compute $p(y_t|x_t^r, x_t^o)$ with Equation 3.2 or 3.3 given the observation y_t

Update weights: $w_i = w_i\, p(y_t|x_t^r, x_t^o)$

Compute total weight from updated w_i: $\eta = \sum_{i=1}^{N} w_i$

Normalize weights: $w_i = w_i \eta^{-1}$

Construct the CDF: $c_i = c_{i-1} + w_i$

for j in X_t **do**

  $s = \text{Uniform}(0, 1)$

  if $c_j > s$ **then**

    Append to X_{t+1}: x_j, w_j

  end if

end for

Compute total weight from $w_i \in X_{t+1}$: $\eta = \sum_{i=1}^{N} w_i$

Normalize weights $w_i \in X_{t+1}$: $w_i = w_i \eta^{-1}$

return X_{t+1}

Though the above approximation is discrete, the same algorithm with minor adjustment can be utilized towards a *regularized particle filter*. A regularized particle filter yields a continuous approximation by using a symmetric PDF as the regularization kernel, centered at each particle, in place of the Dirac delta function [40]. As resampling from a discrete distribution in the SIR filter can lead to a loss of diversity among particles, known as sample impoverishment, the regularized filter resamples from the continuous approximation. Such an approximation is given generally as

$$m(x^o) = \sum_{i=1}^{N} w_i\, K(x^o; x_i)$$

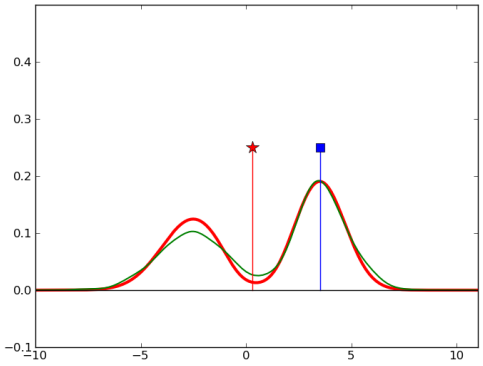
where K denotes the symmetric PDF and the previously imposed constraint, given in Section 3.3.1, is placed upon the weights. An obvious choice for the kernel is a Gaussian density with a mean at each particle given by the resampling stage. That is,

$$m(x^o) = \sum_{i=1}^{N} w_i\, G(x^r; x_i, \Sigma)$$

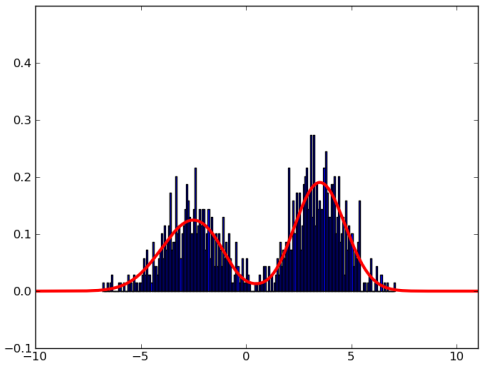
which is simply a finite mixture model of Gaussians. Even though the general particle filtering technique is indeed non-parametric, the use of Gaussian kernels yields a continuous, finite approximation of the original PDF. This is an advantage over a generalized Bayesian filtering problem which first represents the belief as particles and then incorporates the expectation-maximization (EM) method for a continuous approximation representation [41]. Though it has been shown that the E-step can be rewritten using Bayes' theorem [42], this approach has only been given for standard Gaussian mixture models. However, EM for atypical Gaussians mixtures does exist [38], but suffers from the poor scalability of k-means clustering and is only demonstrated for mixture models comprised of very few components. For these reasons and ease of extension, the regularized SIR particle filter is a desirable approximation technique.

Figure 3.6 contrasts the exact exponential family representation with the regularized particle filter approximation. Utilizing this continuous approximation approach allows the extension of the greedy gradient ascent search strategy, where Equation 3.23 is now simply in terms of only N positive Gaussian components. As the Gaussian distribution is indeed a member of the exponential family class, note that the general update rules in Section 3.3.1 still hold.

This concludes Chapter 3 of the thesis. As the complexity of the exact object map representation may not make it feasible for determining "how" to search, the particle filtering approximation produces a finite representation suitable for such an approach. The next section presents experimental results of the greedy search strategy given the approximation using the regularized SIR particle filter.



(a) $t = 1, y_1 = 0$



(b) $t = 1, y_1 = 0$

Figure 3.6: Regularized particle filter using a Gaussian kernel (top) for $\mathcal{W} = \mathbb{R}^1$. The thick, red line denotes the exact object map, and the thin, green line is the approximation. The bottom figure is a histogram of the $1,000$ samples obtained before approximation.

4 EXPERIMENTAL RESULTS

This chapter presents the results of the object map approximation discussed in Section 3.4, asserting the validity of the proposed Bayesian model. Both single agent and multi-agent search simulations are presented, where each employs the gradient ascent strategy. These simulations were accomplished using the programming language Python [43].

4.1 Single Agent Search

First note that the map size, i.e. the area where the search is focused, follows from previous work. Such work dictates that it is a scaled representation of the search space [19, 21, 23, 24]. However, the model presented in this thesis is readily extensible towards larger search spaces because the model is given over the entire workspace. The simulation environment is simply meant to illustrate the model and search process concisely.

Here the search problem is simulated using a single agent to locate an object. The gradient ascent step size that was introduced in Equation 2.2 is specified as $\gamma = 0.1$ for the initial object map. The initial map is given in Figure 4.1 as a mixture of circular Gaussians. Again the black star denotes the robot's position, and the black square is the object's true position in the workspace. An object is considered "found" if it lies within a specified radius of the robot, where this radius is one-half of the gradient step size. Also, the number of particles used for the regularized particle filter is $N = 600$. The equipped binary sensor's covariance matrix is given by Equation 3.25 with $\sigma_s^2 = 0.40$, which is used in all simulations presented. As a "tight" sensor variance implies that the sensor has a low likelihood of firing when the robot is far from the object, approximation of the object map is necessary due to the exponential growth of the exact mixture components.

First, the behavior of the sensor model is discussed. Given the initial object map in Figure 4.1, Figure 4.2 illustrates the gradient ascent strategy dictating

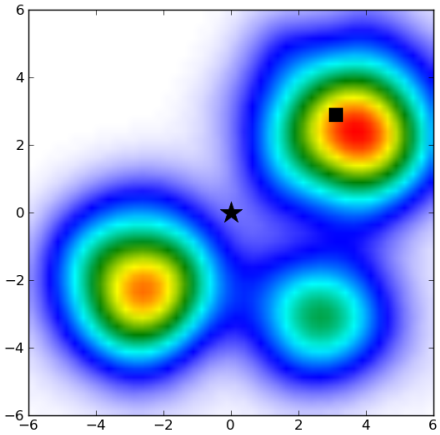
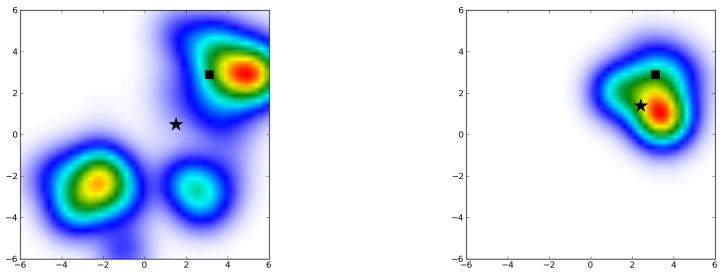


Figure 4.1: The initial object map represents three possible hypotheses for the object's location.

the robot's motion. At time-step $t = 15$, the robot's trajectory has shifted the belief towards the more likely areas, because all sensor observations have yielded negative detections. The interesting behavior occurs at time-step $t = 24$: this is the first positive sensor observation.

Due to this observation, as previously noted in Section 2.1, the belief closest to the robot is scaled up greatly as a consequence of the derived update laws. This can be further seen at time-step $t = 28$ where the successive positive detections localize the remaining probability away from the object. However, for this trial of search, before the robot reaches the peak of the object map, a combination of positive and negative detections is observed. Such is the value of a small sensor variance because an increasing distance from the true object position results in a lower likelihood of observing positive detections. Thus, the belief is shifted towards the object as seen in time-step $t = 52$. This trial successfully found the object in a total of 57 time-steps

For this map 400 trials were conducted, each using the same initial robot state, sensor variance, prior distribution, and the true object position. For these 400 trials, the average time to find the object was ≈ 1193 time-steps. Intuitively, this is due to the poor nature of the gradient ascent strategy because it requires a random walk to escape the attraction of the peaks localized near the true object's position. If the object's position was exactly known, for this scenario, the robot would need no more than 16 time-steps to reach it. Thus, it can be easily inferred from the sample mean that because



(a) $t = 15, y_{15} = 0$ (b) $t = 24, y_{24} = 1$

(c) $t = 28, y_{28} = 1$ (d) $t = 52, y_{52} = 1$

Figure 4.2: Evolution of the object map following the gradient ascent strategy. At time-step $t = 15$ shows the shift in belief given by successive negative observations. Time-steps $t = 24$ and greater show the likelihood localization due to positive detections.

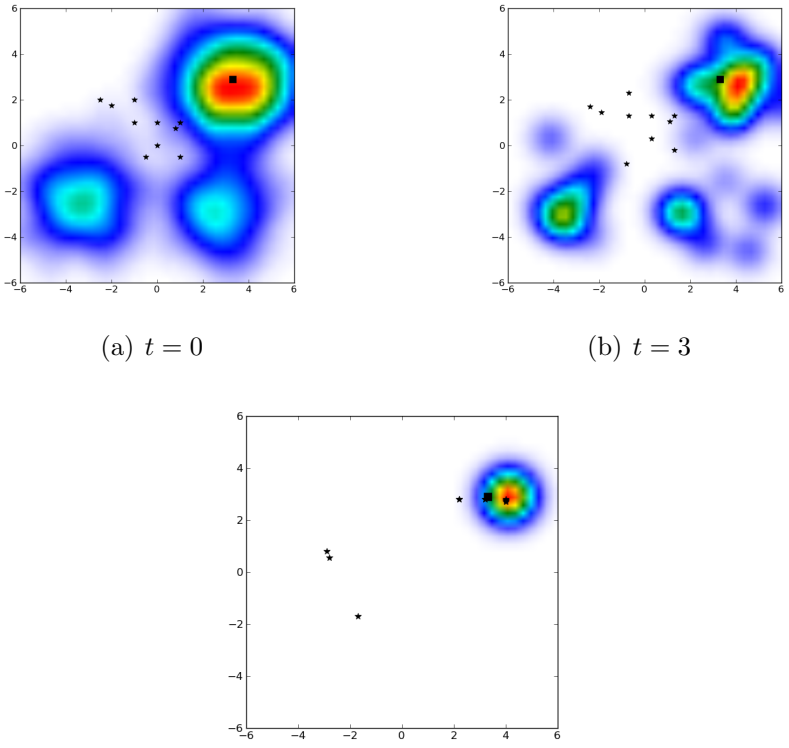
poor localization results in the necessity of a random walk, it is by this random trajectory that the object is eventually found.

4.2 Search with Multiple Agents

The multiple agent process utilized 10 robots for the same initial prior and parameters for the single agent simulations. Each robot is separately governed by the gradient ascent strategy, and the object map is updated as discussed at the end of Section 2.2.2. Specifically, given the object map $m_{t-1}(x^o)$, each robot's current position, and the set of observations $\boldsymbol{Y}_t$, the object map $m_t(x^o)$ is generated by the update laws given in Section 3.3.2. Then, using the current position of an individual robot $x_t^{r,i}$, the i^{th} robot's control input u_t^i is determined by Equations 2.3 and 3.22.

Figure 4.3 shows the advantage of employing multiple agents because from

time-steps one to three all agents observe negative detections. This is noticeable because the belief is scaled down from the occupied agent positions towards the more likely (higher weighted) areas. However, as seen in time-step $t = 32$, the positive detections given by the agents at the peak of the belief dominate the negative detection of the single agent (directly to the left of the remaining probability mass). Though not completely visible, there is one agent in range of the object such that detection occurs during the next time-step. As each agent moves according to the direction of the gradient independently, each must also separately determine when to perform a random walk. One case is obvious: if the control input is zero (hence, the gradient is zero). However, if the ascent step size is too large, i.e. $u_t = 0$ may not occur, the robot can compare its previous positions to ascertain if it is indeed trapped. When following such a strategy, the increased number of searchers cover more area and provide more information about the object



(a) $t = 0$

(b) $t = 3$

(c) $t = 32$

Figure 4.3: Multiple robots utilizing the gradient ascent strategy at three timesteps. The stars denoting the robot positions are smaller for the sake of presentation.

in the workspace. Similarly, having multiple agents invoke the random walk allows farther exploration of the environment beyond the capabilities of a single searcher. For the multi-agent simulations, 400 trials were performed in the manner similar to the single agent results. The average time using multiple agents was ≈ 89 time-steps. This result affirms the intuitive benefits previously discussed, which indeed illustrates the advantage of employing multiple agents for the greedy strategy.

This closes the discussion of the simulations and their results. The next chapter concludes the thesis, and summarizes the presented work. Finally, future work is discussed that is relevant to the search model presented.

5 CONCLUSION

This chapter first summarizes the problem described in this thesis and the results obtained. A number of problems related to the single stationary target search problem are also discussed. These related problems are also readily extensible to other search tasks, e.g. non-stationary targets, that fit the model given in the thesis. Finally, other approximation methods are presented that complement the approach of using the exponential family class.

5.1 Summary

This thesis considered the problem of modeling search for a stationary object in a continuous workspace using a binary sensor. An overview of the Bayesian filter was presented in terms of this problem. The general Bayesian filtering (or inference) equations were derived. These equations model the posterior likelihood of the object's location, which is conditioned on the robot's motion and a sequence of binary sensor observations. A sub-optimal search strategy was given for the robot to determine the object's true position given its belief. Furthermore, the search model and strategy were extended towards the use of multiple search agents.

Though many generalized models exist that strictly utilize Gaussian mixture models, the overarching update laws given result in a target belief encompassing the class of exponential families. An overview of the properties of the exponential family class was given, noting the importance of self-conjugacy. For the defined search problem, an exact, finite representation of the object map was explicitly shown when the likelihood is also a member of the class of exponential families. This was proven given a specific class of initial conditions and employing the exponential family self-conjugacy property. The specifics of this likelihood, i.e. the sensor model was given, and the effect of an increasing sensor variance were discussed.

Due to the increasing number of components in the exponential family mixture model, the Monte Carlo approximation method of particle filtering was utilized. This approximation allowed a static, finite number of components after each update step, which was highly desirable compared to the possibly exponential component growth of the exact representation. Furthermore, the utilization of a symmetric, continuous kernel for the regularized particle filter yielded a continuous approximation in the form of a mixture of Gaussians. Finally, the use of the gradient-ascent search strategy validated the Bayesian model and approximation for this search problem.

A number of target applications forms the basis for future exploration of this approach. The next section gives an overview of these applications and also describes alternative approximation techniques.

5.2 Future Work

Since the viability of the Bayesian search model with a binary sensor has been ascertained, there are various relevant extensions of this work. The use of binary observations possibly allows the generalization of richer sensing platforms, e.g. a camera with a feature detector for faces or vehicles. However, the sensor model is not strictly limited to a binary sensor, since camera and range sensor models can be modeled with the Gaussian distribution. Such a sensor model is sufficient because the self-conjugacy property of the exponential family Bayesian update and the update law presented are valid. Though this thesis pertains to the problem of modeling search, various search strategies in addition to the gradient ascent method indeed exist, and current research presented in Section1.2 compares such methods. Thus, these strategies can be employed and evaluated in the model presented.

The non-stationary single target search model described in this thesis can also be applied to a moving target. This extension is simply a search-and-track problem where the primary objective is to find a moving target and, upon detection, successfully track said target for the duration of the search mission. Again, as previous and recent work has illustrated, the general Bayesian approach given encapsulates the belief representation within the class of exponential families. Another area that can be addressed is target localization with a single search agent. Such a problem could entail the use of

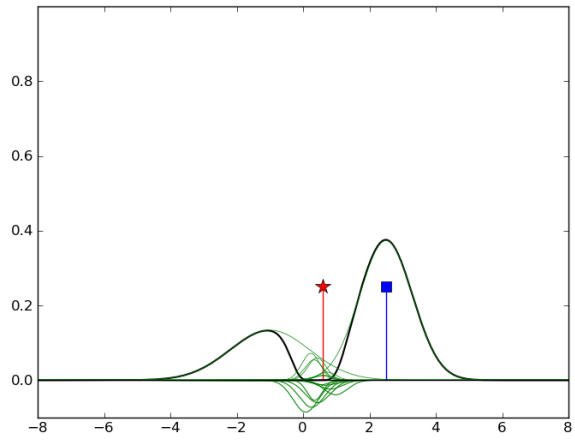
(solely) a binary sensor to determine the location of an object. However, if the observation model is probabilistic, the object cannot be perfectly localized in finite time. Thus, the criterion to be minimized would be the expected entropy of the object map. Additionally, a policy that guarantees consistent estimation is necessary for the search-and-track objective.

Furthermore, the presented model is readily extensible to the multiple target search problem. Affirming that all targets are independent of each other allows their belief representation to be partitioned into separate probability maps, which is feasible for planning a search strategy. Again, as in the single target case, the problem model given in this thesis provides straightforward evolution rules for a binary sensor. The overall goal would be to maximize the number of target detections or to minimize the expected time for each target individually. Additionally, employing multiple agents can be extended to the search for multiple targets where each target is described singly with a shared belief.

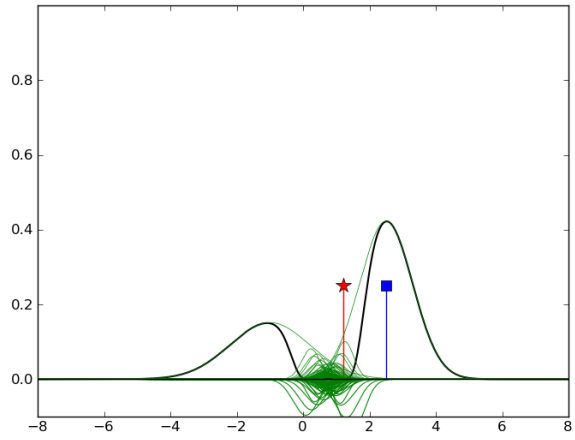
One advantage of using mixtures of distributions from an exponential family is that many good approximation techniques exist. For example, algorithms for soft [44] and hard [45] clustering based on Bregman divergences can significantly reduce the number of components required in the representation while causing only a small perturbation (with respect to relative entropy) between the true and approximated distributions. These clustering techniques not only show that the Bregman divergence is equivalent to the Kullback-Leibler divergence but yield entropy and centroid calculations solely based upon the exponential family canonical parameterization. The entropy is defined in terms of the log normalizer, which is a function whose arguments are the natural parameters. To calculate the centroids, the natural parameters for each component are considered. Furthermore, mixtures from the process model described in this paper tend to be good candidates for approximation using these methods because the different components tend naturally to form clusters.

The formation of component clusters is due to the incremental updating of the map and robot position. Consider Figure 5.1 which shows a snapshot of the time evolution of a one-dimensional map with an initial distribution based on a mixture of two Gaussian distributions. In these figures, the thick black curve indicates the function $m_t(x)$, and the thin green curves are the individual components comprising the mixture model. From the figure, one

can see that, in this case, the components form well-defined clusters as they increase. Experimental experience of the exact belief representation tends to indicate this is not an exceptional case.



(a) $t = 3, y_3 = 0$



(b) $t = 6, y_6 = 0$

Figure 5.1: Exact evolution of the object map for $\mathcal{W} = \mathbb{R}^1$. The means, or peaks, of the components (thin green lines) can be easily "grouped" by sight.

REFERENCES

[1] H. R. Richardson and J. H. Discenza, "The United States Coast Guard computer-assisted search planning system (CASP)," *Naval Research Logistics Quarterly*, vol. 27, no. 4, pp. 659–680, 1980.

[2] B. O. Koopman, "Search and its optimization," *The American Mathematical Monthly*, vol. 86, no. 7, pp. 527–540, 1979.

[3] H. R. Richardson and L. D. Stone, "Operations analysis during the underwater search for scorpions," *Naval Research Logistics Quarterly*, vol. 18, no. 2, pp. 141–157, 1971.

[4] L. Stone, *Theory of Optimal Search*, ser. Mathematics in Science and Engineering. New York, NY: Academic Press, 1975, vol. 118.

[5] Y. Zhang, M. Schervish, and H. Choset, "Probabilistic hierarchical spatial model for mine locations and its application in robotic landmine search," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 681–689.

[6] B. Koopman, *Search and Screening*. Elmsford, NY: Pergamon Press, 1980.

[7] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, "A survey of the search theory literature," *Naval Research Logistics (NRL)*, vol. 38, no. 4, pp. 469–494, 1991.

[8] L. D. Stone, "What's happened in search theory since the 1975 Lanchester Prize?" *Operations Research*, vol. 37, no. 3, pp. pp. 501–506, 1989.

[9] D. W. Gage, "Randomized search strategies with imperfect sensors," in *Proceedings of SPIE Mobile Robots VIII*, 1993, pp. 270–279.

[10] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 2, pp. 99–110, 2006.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[12] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding, "A Bayesian Algorithm for simultaneous localization and map building," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics. Berlin, Germany: Springer, 2003, vol. 6, pp. 49–60.

[13] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY: Academic Press, 1970.

[14] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, June 2005.

[15] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[16] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[17] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, May 2011.

[18] E. Sommerlade, B. Benfold, and I. Reid, "Gaze directed camera control for face image acquisition," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, May 2011.

[19] T. Furukawa, F. Bourgault, B. Lavis, and H. Durrant-Whyte, "Recursive Bayesian search-and-tracking using coordinated uavs for lost targets," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, May 2006, pp. 2521–2526.

[20] H. Choset, "Coverage for robotics-A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[21] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Berlin, Germany: Springer, 2006, vol. 24, pp. 209–222.

[22] E.-M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 3169–3174.

[23] T. Chung, "On probabilistic search decisions under searcher motion constraints," in *Algorithmic Foundation of Robotics VIII*, ser. Springer Tracts in Advanced Robotics. Berlin, Germany: Springer, 2009, vol. 57, pp. 501–516.

[24] T. Chung and J. Burdick, "A decision-making framework for control strategies in probabilistic search," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4386–4393.

[25] T. H. Chung and J. W. Burdick, "Multi-agent probabilistic search in a sequential decision-theoretic framework," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 146–151.

[26] T. H. Chung and S. Carpin, "Multiscale search using probabilistic quadtrees," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[27] S. Ross, *First Course in Probability, A*, 8th ed. Upper Saddle River, NJ: Prentice Hall, Nov. 2008.

[28] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: John Wiley and Sons, Inc., 2006.

[29] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[30] G. E. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*. New York, NY: Wiley-Interscience, 1992.

[31] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[32] B. O. Koopman, "On distributions admitting a sufficient statistic," *Transactions of the American Mathematical Society*, vol. 39, no. 3, pp. pp. 399–409, 1936.

[33] G. Darmois, "Sur les lois de probabilit estimation exhaustive," *Comptes Rendus de l'Acadmie des Sciences*, vol. 200, pp. 1265–1266, 1935.

[34] E. J. G. Pitman, "Sufficient statistics and intrinsic accuracy," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 32, no. 4, pp. 567–579, 1936.

[35] L. Brown, *Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory*. Hayward, CA: Institute of Mathematical Statistics, 1986.

[36] P. Diaconis and D. Ylvisaker, "Conjugate priors for exponential families," *The Annals of Statistics*, vol. 7, no. 2, pp. 269–281, 1979.

[37] F. Nielsen and V. Garcia, "Statistical exponential families: a digest with flash cards," arXiv:0911.4863, Nov. 2009.

[38] B. Zhang and C. Zhang, "Finite mixture models with negative components," in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner and A. Imiya, Eds. Berlin, Germany: Springer, 2005, vol. 3587, pp. 31–41.

[39] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, Sept. 2003.

[40] A. Doucet, N. D. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer, 2001.

[41] P. Pfaff, C. Plagemann, and W. Burgard, "Gaussian mixture models for probabilistic localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 467–472.

[42] B. Upcroft, S. Kumar, M. Ridley, L. L. Ong, and H. Durrant-Whyte, "Fast re-parameterisation of Gaussian mixture models for robotics applications," in *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra, ACT, Australia, December 2004, pp. 1–7.

[43] Python Software Foundation, "Python programming language," 2011. [Online]. Available: http://www.python.org

[44] V. Garcia and F. Nielsen, "Simplification and hierarchical representations of mixtures of exponential families," *Signal Processing*, vol. 90, no. 12, pp. 3197–3212, 2010.

[45] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *The Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.